US Army Corps
of Engineers
Construction Engineering
Research Laboratories

USACERL Technical Report EC-94/12
September 1994

Estimating Attenuation and Propagation of Noise Bands From a Distant Source Using the Lookup Program and Data Base

by
Michael J. White

Unavoidable noise generated by military activities can disturb the surrounding community and become a source of complaint. Military planners must quickly and accurately predict noise levels at distant points from various sound sources to manage noisy operations on a daily basis. This study developed the Lookup computer program and data base to provide rapid estimates of outdoor noise levels from a variety of sound sources. Lookup accesses a data base of archived results (requiring about 5 MB disk space) from typical situations rather than performing fresh calculations for each consultation. Initial timing tests show that Lookup can predict the sound levels from a noise source at distances up to 20 km in 1 second on a DOS-compatible personal computer (PC). This report includes the Lookup program source code, and describes the required input for the program, the contents of the archival data base, and the program output. Lookup was written to compile with MS-Fortran, and will run under DOS on any IBM compatible with 640k random access memory. Lookup also conforms to ANSI 1978 standard Fortran and will run under the Unix operating system.

USER EVALUATION OF REPORT

REFERENCE: USACERL Technical Report (TR) EC-94/12, *Estimating Attenuation and Propagation of Noise Bands From a Distant Source Using the Lookup Program and Data Base*

Please take a few minutes to answer the questions below, tear out this sheet, and return it to USACERL. As user of this report, your customer comments will provide USACERL with information essential for improving future reports.

1. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

____

____

____

2. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.)

____

____

3. Has the information in this report led to any quantitative savings as far as manhours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

____

____

4. What is your evaluation of this report in the following areas?

a. Presentation:____

b. Completeness:____

c. Easy to Understand:______________________________________

d. Easy to Implement:_______________________________________

e. Adequate Reference Material:_______________________________

f. Relates to Area of Interest:_________________________________

g. Did the report meet your expectations?________________________

h. Does the report raise unanswered questions?_____________________

i. General Comments. (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.)

____

____

____

____

____

5. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: ____________________

Telephone Number: ____________________

Organization Address: ____________________

____

____

6. Please mail the completed form to:

Department of the Army
CONSTRUCTION ENGINEERING RESEARCH LABORATORIES
ATTN: CECER-IMT
P.O. Box 9005
Champaign, IL 61826-9005

Notice to Program Recipients

The program included with this document is furnished by the U.S. Government and is accepted and used by the recipient with the express understanding that the Government makes no warranty, expressed or implied, concerning the accuracy, completeness, reliability, usability, or suitability for any particular purpose of the information and data contained in this program or furnished in connection therewith, and the United States shall be under no liability whatsoever to any person by reason of any use made thereof.

The program belongs to the Government. Therefore, the recipient further agrees not to assert any proprietary rights therein or to represent this program to anyone as other than a Government program. The recipient also agrees that the program and all documents related thereto, including all copies and versions (except when expressly authorized otherwise) in possession thereof, will be discontinued from use or destroyed upon request by the Government.

This program may be copied without restriction, and is to be used only in the public interest and/or the advancement of science and will not be used by the recipient to gain unfair advantage over any client or competitor. Whereas the recipient may charge clients for the ordinary costs of applying the program, the recipient agrees not to levy a charge, royalty, or proprietary usage fee (except to cover any normal copying and/or distribution costs) upon any client for the development or use of the received program. Recipients desiring to modify and remarket the program will be required to comply with a separate agreement. Only minor or temporary modifications will be made to the program (e.g., necessary corrections or changes in the format of input or output) without written approval from the Government. Should the program be furnished by the recipient to a third party, the recipient is responsible to that third party for any support and upkeep of the program. Information on the source of the program will be furnished to anyone requesting such information.

The accuracy of this program depends entirely on user-supplied data. It is the user's responsibility to understand how the input data affects the program output and to use the output data only as intended.

All documents and reports conveying information obtained as a result of the use of the program by the recipient will acknowledge the Corps of Engineers, Department of the Army, as the origin of the program. All such documentation will state the name and version of the program used by the recipient.

No support group has been established for this program. The Government assumes no responsibility for support, maintenance, or update of the existing software or documentation.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1994	Final

4. TITLE AND SUBTITLE
Estimating Attenuation and Propagation of Noise Bands From a Distant Source Using the Lookup Program and Data Base

5. FUNDING NUMBERS
4A162720
A896
AI-T53

6. AUTHOR(S)
Michael J. White

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Construction Engineering Research Laboratories (USACERL)
P.O. Box 9005
Champaign, IL 61826-9005

8. PERFORMING ORGANIZATION REPORT NUMBER
EC-94/12

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Headquarters, Department of the Army (HQDA)
ATTN: ACSIM
600 Army Pentagon
Washington, DC 20310-0600

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
Unavoidable noise generated by military activities can disturb the surrounding community and become a source of complaint. Military planners must quickly and accurately predict noise levels at distant points from various sound sources to manage noisy operations on a daily basis. This study developed the Lookup computer program and data base to provide rapid estimates of outdoor noise levels from a variety of sound sources. Lookup accesses a data base of archived results (requiring about 5 MB disk space) from typical situations rather than performing fresh calculations for each consultation. Initial timing tests show that Lookup can predict the sound levels from a noise source at distances up to 20 km in 1 second on a DOS-compatible personal computer (PC). This report includes the Lookup program source code, and describes the required input for the program, the contents of the archival data base, and the program output. Lookup was written to compile with MS-Fortran, and will run under DOS on any IBM compatible with 640k random access memory. Lookup also conforms to ANSI 1978 standard Fortran and will run under the Unix operating system.

14. SUBJECT TERMS
sound propagation prediction
noise levels
Lookup Program
archival data base
military training

15. NUMBER OF PAGES
42

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

FOREWORD

This work was performed for the Office of the Assistant Chief of Staff for Installation Management (ACSIM), Headquarters, U.S. Department of the Army (HQDA) under Project 4A162720A896, "Environmental Quality Technology"; Work Unit AI-T53, "Weather-Based Blast Noise Prediction." The technical monitor was Timothy Julius, DAIM-EO.

The research was done by the Environmental Compliance Modeling and Simulation Division (EC), Environmental Sustainment Laboratory (EL), U.S. Army Construction Engineering Research Laboratories (USACERL). The USACERL principal investigator was Dr. Michael White. Dr. William Goran is Acting Chief, CECER-EC, and Dr. Edward Novak is Acting Chief, CECER-EL. The USACERL technical editor was William J. Wolfe, Information Management Office.

LTC David J. Rehbein is Commander and Acting Director of USACERL, and Dr. Michael J. O'Connor is Technical Director.

CONTENTS

ESTIMATING ATTENUATION AND PROPAGATION OF NOISE BANDS FROM A DISTANT SOURCE USING THE LOOKUP PROGRAM AND DATA BASE

1 INTRODUCTION

Background

Unavoidable noise generated by military training activities can disturb the surrounding community and become a source of complaint. Military planners must be able to quickly and accurately predict noise levels at distant points from various sound sources to manage noisy operations on a day-to-day basis. Predicting sound propagation outdoors for broadband noise sources or for long distances can require considerable computational effort. A number of available research-grade computer codes can predict sound propagation from outdoor sources, all of which involve some kind of limitation (in accuracy or model assumptions) or expense that make them impractical for routine use. For example, the Fast Field Program and the Parabolic Equation methods give very precise solutions to the wave equation, but with a tremendous expense in central processor unit (CPU) time.

In fact, such expensive or inefficient programs may not be necessary for routine sound propagation prediction. Even with these full-wave acoustical models, there is still a need to ensure the accuracy of calculated results by comparing them with large amounts of measured propagation data. Moreover, many situations in outdoor propagation studies recur. For example, the micro-climate near the ground can show diurnal cycles whose sound speed profiles assume "typical" shapes and thus produce similar propagation; the listeners are near the ground; the sources are stationary or slow-moving when compared to the speed of sound, and so on. It may be more practical to archive results from typical situations and access them through computational lookup tables, rather than to specify new parameters and do fresh calculations for each consultation.

Objective

The objective of this work was to devise a method to predict sound propagation of noise levels resulting from military training activities, with the accuracy of full-wave acoustical models, in near-real time.

Approach

Many thousands of propagation cases were calculated, and the results were organized and tabulated for easy access. The many parameters that influence propagation were systematically varied to approximately cover most situations; i.e., the source height, receiver height, receiver range, frequency, wind speed and direction, temperature gradient, and ground impedance were varied in successive computations to build a data base of typical results.

Two computer programs were written to access the lookup table and produce estimates of the received sound level. The Lookup program accepts an input file that describes the acoustical source and environment of interest, and predicts the sound spectrum and frequency-weighted sound levels at selected locations. This prediction is formed from the best-match case in the lookup table, which is adjusted to account for differences between the stored and desired cases. For example, the correction for attenuation due to molecular absorption is such an adjustment. Sound Propagation (SP) is the second program, which assembles information about

the propagation problem at hand, displays and manipulates it graphically, prepares input for Lookup, and retrieves and displays the predictions.

Note that Lookup is a user-*unfriendly* program that deals only with the mechanics of the calculations and the efficient access to the tables. (It runs a typical case in under 1 second on the PC.) SP is the user-*friendly* graphical user interface that makes it easy to enter, display, and edit cases of interest, and quickly view the results.

Technical Requirements

Lookup was written to compile with MS-Fortran,* and will run under DOS on any IBM compatible with 640 kB random access memory (RAM). Lookup also conforms to ANSI 1978 standard Fortran and will run under the Unix operating system. The Lookup data base requires about 5 MB of free disk space.

Mode of Technology Transfer

The Lookup program and data base will be forwarded to the U.S. Army Environmental Hygiene Agency (AEHA) for fielding in the "Weather-Based Noise Now-Casting System."

---

*MS-Fortran is a product of Microsoft Corp., Redmond, WA.

2 The Lookup Program

Purpose

The purpose of the Lookup program is to provide a rapid estimate of sound levels at a distance from the sound source using precalculated tables, without incurring the expense of a lengthy specific calculation for the problem at hand. Source code for Lookup is included in Appendix A.

Lookup Input

The Lookup input file contains all of the physical parameters needed to specify an acoustic propagation problem, such as the source and receiver heights, wind speed and direction, type of ground surface, etc. Also included in the file is the name of the Lookup table data base directory and the name of the index file for the table. Appendix B shows a sample Lookup input file, which is composed of a strict format, for ease in checking input errors. This input format is fairly easy for the user or the computer to read, but more difficult for the user to write. Each major entry or group of entries is preceded by a header line describing the entry. These headers are required by the Lookup program, and most of the entries are required, although some are of variable length.

Version Number

The first header and entry specify the input file version ID number. This number ensures that any future changes in the input file format can be recognized by corresponding upgrades of the Lookup program.

Directory Name

The next header and entry pair contain the directory name for the excess levels database and index file. The directory name should be chosen to be the greatest common path name for the files in the data base. Note that the data base may extend over several directories, but the directory name should be common to all of the data files. For example, as initially distributed, the Lookup data base has four directories:

```
C:\LOOKUP\SRCA
C:\LOOKUP\SRCB
C:\LOOKUP\SRCC
C:\LOOKUP\SRCD
```

the Lookup index file name is:

```
C:\LOOKUP\INDEX10.FIL
```

and typical entries in the index file are:

```
SRCA\ASGP10.AVE ...
SRCC\CHGP10.AVE ...
SRCD\DSGP10.AVE ...
```

The directory name sent to Lookup should read:

C:\LOOKUP\

The directory name should start on the first character position on the line following the header.

Index File Name

The next header and entry include the Lookup index file name for the excess levels table. Appendix C shows an excerpt from the index file. The index file contains a line-by-line listing of the file names of all of the files in the data base, paired with the propagation parameters for each type of case examined. The parameters include the source and receiver heights, type of ground surface, etc. The name of the index file and the file names listed in the index file should be specified relative to the common directory name, as above. The index file name should be printed starting with the first position on the line following the header. In the example from Appendix A, the index file name is "INDEX10.FIL".

Source Height

The source height is the height of the source above the ground surface (in meters). If the ground is rugged or sloping, the source height used is the distance along a perpendicular from a local average surface plane. The initial release of Lookup cannot form predictions for buried sources, and thus will not accept negative source heights. If the source falls below the average local surface height, but is still "above" ground, the source height should be set equal to zero. Negative source heights will cause Lookup to halt.

Receiver Height

All conditions that apply to the source height, apply similarly to the receiver height.

Receiver Range

The horizontal distance from source to receiver is the "receiver range" and must be entered in meters. If zero distance is given as input, a calculation will be made for each range in the data base, for a total of 44 distances. At each distance, only the overall levels will be reported, and no spectral information will be provided. If a positive distance is specified instead, a different set of information will be given on output. In this case, only a calculation for that range will be performed. Also, an estimate of the receiver 1/3-octave spectrum will be reported in addition to the overall levels at that distance. A negative receiver range will cause Lookup to halt. For the format of the output, see the section, "Lookup Output." Also see "Method of Calculation" for information on the behavior of Lookup with regard to receiver range.

Receiver Azimuth

The header and entry for receiver azimuth gives the direction of the receiver from the sound source. The units of azimuth are degrees, and are to be measured from zero degrees north. Units of azimuth should increase with clockwise rotation, looking down on the site from above. The receiver azimuth is only used in computing the wind component in the direction of propagation, but must be included in the input file, even if the wind profile is unknown or unspecified.

Ground Classification

The next entry is the porosity classification of the surface. In the Lookup data base the ground has been classified on a continuous scale from hard (class 0) to porous (class 1) as the fraction of porous ground between the source and receiver. Note that not all types of ground surfaces should be classified in this way (e.g., snow-covered ground should not), but this scheme could be expanded for more types of surfaces by adding parameters. Ground classification values outside of the interval (0,1) will cause Lookup to halt.

Ground Roughness Height

The next entry is the estimate of the roughness (in meters) of the ground surface (rms). The ground roughness is used only in estimating the thickness of the roughness layer of the wind profile. Normally, the minimum roughness height for very flat ground is about 0.1 m. Nonpositive roughness heights will cause Lookup to halt.

Temperature Profile

The variation of temperature with height is call the temperature profile, which is to be supplied in a list under the temperature profile header. On each line, the height of the sensor should precede the temperature measurement. The sensor height should be supplied in meters, and the temperature in degrees centigrade. Temperatures outside of the interval (-50 °C, 50 °C) will cause Lookup to halt. A negative height and arbitrary temperature should be given to signal the end of the temperature profile list. Note: only the first 100 height and temperature pairs are used by Lookup; the rest are simply discarded. Default values for the temperature list are substituted if there are too few values to construct a temperature gradient over height.

Humidity Profile

The next entry in the input file is the humidity profile as a function of height. The profile should be supplied as in the temperature profile entry, with the height first, and the relative humidity, following on the same line. The relative humidity should be entered in units of percent. As for the temperature profile, only the first 100 entries of the humidity profile will be used; the rest are discarded.

Values of percent relative humidity outside of the interval (1,100) will cause Lookup to halt. The molecular absorption routine is less accurate at very low humidities, and excluding the interval (0,1) helps detect errors that might occur when relative humidity has not been properly converted to a percentage.

Wind Profile

The wind profile is entered below the humidity profile and follows a similar format. The sensor height, wind speed, and wind direction for each wind sensor are listed on each line of the profile. The profile list should end with an entry of negative height and with arbitrary (nonempty) values for wind speed and direction. The wind speed and direction are expected to be expressed in units of meters/second and degrees from north, respectively. The wind direction should be entered in the input as the meteorological wind direction, i.e., the direction the wind vane points.

Source Level Spectrum

The source spectrum should appear below the meteorological profiles. The source spectrum should be entered as 1/3-octave band levels (dB) starting with band number zero (1 Hz) and ending with band

number 43 (20 kHz). There must appear values for each band index. Unused bands should be set equal to -100.0 dB.

To obtain only the spectral decay from the source, set all of the bands equal to zero dB, the reference distance (described below) to 1 m, and specify the range of interest (rather than letting range equal zero).

The input spectrum should be flat-weighted, although Lookup will operate with and perform calculations on a frequency-weighted spectrum. It should be understood that a frequency-weighted input spectrum will not obtain proper values for the A- and C-weighted overall levels. The output spectra will have the same frequency weighting as that applied to the input spectrum.

The source spectrum should also be taken from a far-field measurement and taken at a distance at which the nonlinear effects of propagation are weak. For example, measurements should be made about 10 wavelengths from the source, and at sound levels below 160 dB. The effects of the ground reflection are assumed to be removed from the source spectrum.

Reference Distance

The reference distance entry for the source spectrum follows the spectrum in the input file. The distance should correspond to the distance at which the source spectrum was measured.

User Weighting Spectrum

The user weighting curve is applied to the receiver spectrum, and the user-weighted spectrum is summed by Lookup to give an overall user-weighted level. The user weighting spectrum follows the same format as the source spectrum, with some changes. Only inactive (-100.0 dB) source bands can have corresponding unspecified bands in the user curve. All other bands supplied from the user curve are used. There must appear user-curve values for all 1/3-octave bands zero through 43. The user weighting spectrum concludes the Lookup input file.

Lookup Output

The output "file" is written by Lookup to the standard output and may be saved as a file, viewed on the display screen (as numbers and text) or sent to another program for graphical display or for further processing.

The output file will contain an error report if an error occurred during the calculation. If there were no errors, a report is given that describes the Lookup parameters used for searching the excess levels data base and the best match found from the index. Appendix D shows the contents of a sample output file.

The predictions for the overall levels will be printed next. If a particular range is selected (input receiver range positive), the range, overall level, overall A-weighted level, overall C-weighted level, and overall user-weighted level are printed on one line. If all of the ranges are selected (input receiver range zero), one such prediction is made for every range in the excess levels data base (from 1 m to 20 km). If a specific range is selected, the predicted 1/3-octave spectrum at the receiver will be printed.

Lookup Data Base

The excess levels data base is composed of an index file and many files of excess level spectra as a function of distance.

Each excess spectrum file has a list of range numbers and spectrum pairs from some starting range (say 1 m or 10 m) to some final range (perhaps 1 km or 20 km). The range numbers are defined as 10 times the common logarithm of the range. The range numbers advance by one for each spectrum in the list, such that the corresponding ranges advance through the list by a geometric progression in 1/3-octave multiples.

The spectrum associated with each range represents the expected change in level between the source and receiver, due to such effects as ground reflection, ground absorption, refraction, diffraction, and ducted propagation of sound. Effects of refraction due to changes in the atmospheric properties with height are accounted for. The spectra do not contain the effects of spherical spreading and absorption, which are accounted for separately in the Lookup program.

The spectra pertain to 1/3-octave bands of noise, rather than pure tones, although they were constructed from a pure tone propagation model. There is also an implicit assumption that the source spectrum is roughly "flat" within any given band.

The excess levels files were constructed from a pure-tone, full-wave, two-way wave equation solution, using the Fast Field Program (FFP). The FFP was set to run a variety of propagation problems for all of the 1/3-octave band center frequencies. The FFP provides a prediction of the excess level at a number of equally spaced distances between source and receiver.

The predictions from the FFP (or any other wave model) usually vary in level with distance due to wave interference and refraction. (In other words, the levels vary in response to all of the parameters we specify.) In some situations the predictions show considerable variation or oscillation with range. In circumstances that produce strong interference patterns, it is debatable whether the structure has any practical significance; in a perfectly time-invariant and precisely-known environment, perhaps it is significant. In a real environment, not one atmospheric measurement can be made with great precision, nor will any of the parameters remain invariant with time. Furthermore, one seldom encounters sources that produce only pure tones.

As stated, the full wave solutions as a function of range do contain "structure," and some of the rapid variation in range is undesirable. The more significant features of the predictions are those that persist over great distance or those that do not vary rapidly with frequency or from slight changes in the atmosphere.

Within any frequency band, the band power spectrum is the frequency integral of the absolute square of the source and transfer function product. If either the source spectrum or the transfer function is relatively flat within the band, the integral can be simplified by replacing the source spectrum by its rms average. The integral can be further simplified when the oscillating part of the transfer function (oscillations with respect to range) are due to positional changes over an interference pattern. In this case, a similar oscillating pattern can be found by holding the range constant and sweeping through the frequency.

The propagation model provides the signal level at a particular frequency over many ranges, but of interest here is the level at a particular range, averaged over frequency. The average over the frequency interval should be well approximated by the average over the same number of oscillation over the range.

It can be shown that the approximation holds, provided the range interval is large enough to admit many oscillations, or small enough so that the transfer function does not change appreciably. If the range oscillations behave such that the wavelength is equal to the familiar quotient of sound speed and frequency, the range of integration should extend over the 1/3-octave range intervals. The rms average over the 1/3-octave band can be replaced by an average over the 1/3-octave range interval. Note that the wavelength of the oscillations in range may not obey the relation above, but it is not a strict requirement for the argument to hold.

The predictions from the FFP were range-averaged in an attempt to remove the unwanted oscillations in the Lookup data base files. The arithmetic standard deviations of the levels over these range intervals were also calculated and stored. The standard deviation is a useful indicator of the possible variation in levels due to small changes in position, frequency, or environmental conditions. Future improvements to the Lookup program might involve the use of this quantity for a prediction of the anticipated variation in levels, or the probability of exceeding certain thresholds.

Occasionally the FFP yielded predictions at so few ranges that some range intervals contained no prediction. In cases where there are no predicted levels, the data base entries were set equal to 100.0 dB and the corresponding standard deviations were set equal to -1.0 dB.

In the initial release of the data base, several atmospheric and environmental parameters varied in the table. Every combination of the following parameters was used: hard and porous ground; linear and logarithmic sound speed profiles; profile speeds ranging from about -12 m/s to about 12 m/s in steps of 2 m/s; source heights of 2 m, 5 m, 15 m, and 100 m; and frequencies from 1 Hz to 1600 Hz, on 1/3-octave centers. In all, a total of 370 separate cases were prepared for the table.

Though this table does not (cannot) cover all possible situations that might occur in practice, it does provide a starting place for near-ground propagation to medium range (up to about 3 km). Beyond 3 km or so, other influences often become more important, such as hilly terrain or turbulence. The nature of the atmosphere well above conventional surface-based meteorological sensors has a more pronounced effect on levels at longer ranges. Nevertheless, computations for these cases were performed, and stored in the table for distances up to 20 km in range.

3 SUMMARY

This study has created a method to quickly predict sound propagation from outside sources through the Lookup program and data base. Based on the assumption that many situations in outdoor propagation tend to recur, the Lookup program speeds the prediction process by accessing a data base of information describing typical situations, and using that information to estimate received sound levels.

The user provides the Lookup program with an input file that contains the physical parameters of the propagation problem, e.g., the receiver heights, winds speed and direction, type of ground surface, and so forth. The Lookup program estimates sound propagation by matching the entered information against its data base of typical parameters, and doing a calculation based on the closest match.

APPENDIX A: Lookup Program Source Code

```
C-----------------------------------------------------------------------
C     Use lookup table to estimate sound propagation.
C-----------------------------------------------------------------------
      CHARACTER  Dir*80,Best*80,Index*80
      INTEGER  Iband,Prof,Rexs,Rold,Rint
      LOGICAL  Fend,Fopen
      REAL  Srchi,Rechi,Range,Azimth,Grclas,Rufhi
      REAL  Tchi(100),Tc(100),Rhhi(100),Rh(100)
      REAL  Whi(100),Wx(100),Wy(100)
      REAL  TobSrc(0:43),Refdis,Usrcrv(0:43),TobFlt(0:43)
      REAL  Tc0,Dtdh,Rh0,S10,W10,W10pro,Fr,Airab
      REAL  Delair(0:43),Delstp(0:43),Delexs(0:43),Delold(0:43)
      REAL  R,Rng,L,A,C,U,Delspr,Delint(0:43),R2,S,X1,X2
      REAL  TobAwt(0:43),TobCwt(0:43),TobUwt(0:43),Ovrall
C
      EXTERNAL  Airab,Ovrall
C
C     -----Do all the parameter reading.
      CALL Rdver()
      CALL Rdname(Dir,Index)
      CALL Rdgeom(Srchi,Rechi,Range,Azimth)
      CALL Rdgrnd(Grclas,Rufhi)
      CALL Rdtc(Tchi,Tc)
      CALL Rdrh(Rhhi,Rh)
      CALL Rdwswd(Whi,Wx,Wy)
      CALL Rdlvls(TobSrc,Refdis,Usrcrv)
C
C     -----Fit met profiles; choose speed profile; select "best" file.
      CALL Fittc(Tchi,Tc,Tc0,Dtdh)
      CALL Fitrh(Rhhi,Rh,Rh0)
      CALL Fitw(Whi,Wx,Wy,Rufhi,Azimth,W10,W10pro)
      CALL Chprof(Dtdh,W10,W10pro,Prof,S10)
      CALL Fndfil(Prof,S10,Grclas,Srchi,Rechi,Dir,Index,Best)
C
C     -----Molecular absorption spectrum (dB/m) offset from STP.
      DO 10 Iband=0,43
         Fr = 10.0**(Iband/10.0)
         Delair(Iband) = 8.686*Airab(Fr,Tc0,Rh0,1.0)
         Delstp(Iband) = 8.686*Airab(Fr,15.0,70.0,1.0)
 10   CONTINUE
C
C     -----Prepare to read the excess levels file.
      Fopen = .FALSE.
      Fend = .FALSE.
      Rexs = -1
      Rold = -1
C
      WRITE (*,*) 'BEGIN RANGE, FLAT, A, C AND USER'
      DO 70 Rint=0,43
C
C        -----Choose the range number for interpolation or sweep.
         IF (Range.GT.0.0) THEN
            R = 10.0*ALOG10(Range)
         ELSE
            R = FLOAT(Rint)
         ENDIF
C
C        -----Fetch the next good spectrum.
 30      IF (.NOT.Fend.AND.Rexs.LT.R) THEN
C
C           -----Save the old spectrum first.
            Rold = Rexs
```

```
            CALL Hold(Delexs,Delold)
C
C           -----Here we attempt to get the new one (may be EOF, etc.).
            CALL Rdexs(Rexs,Delexs,Fopen,Fend,Dir,Best)
            GOTO 30
C
C        -----Save the good one the last trip through (reached EOF).
         ELSE IF (Fend) THEN
            CALL Hold(Delold,Delexs)
         ENDIF
C
C        -----See if interpolation is possible.
         IF ((Range.GT.0.0).AND.(Rold.GE.0.0).AND.(.NOT.Fend)) THEN
            CALL Ntrpl8(FLOAT(Rold),Delold,FLOAT(Rexs),Delexs,R,Delint)
            CALL Hold(Delint,Delexs)
         ENDIF
C
C        -----Spherical distance.
         Rng = 10.0**(R/10.0)
         R2 = SQRT((Srchi-Rechi)**2+Rng**2)
         Delspr = 20.0*ALOG10(R2/Refdis)
C
C        -----Apply spherical spreading and molec. absorption.
         DO 40 Iband=0,43
C
C           -----Source level.
            S = TobSrc(Iband)
            TobFlt(Iband) = S
C
C           -----Spherical spreading plus air absorb.
            X1 = -Delspr-Delair(Iband)*R2
C
C           -----Remove std. air from table attn.  Cut losses to -25.
            X2 = AMAX1(Delexs(Iband)+Delair(Iband)*R2,-25.0)
C
C           -----Only modify levels if source band is set.  Stop at -100.
            IF (S.GT.-100.0) TobFlt(Iband) = AMAX1(S+X1+X2,-100.0)
C
 40      CONTINUE

C        -----Apply flat (none), A, C, and User weighting.
         CALL Frwtng(TobFlt,TobAwt,TobCwt,TobUwt,Usrcrv)
         L = Ovrall(TobFlt)
         A = Ovrall(TobAwt)
         C = Ovrall(TobCwt)
         U = Ovrall(TobUwt)
C
C        -----Output the Range, Sum, A, C, and User SEL dB.
         WRITE (*,50) Rng,L,A,C,U
 50      FORMAT(1X,F7.1,4(1X,F6.1))
C
C        -----Print the interpolated spectrum.
         IF (Range.GT.0.0) THEN
            WRITE (*,*) 'RECEIVER SPECTRUM, 1/3-OCTAVE BANDS 0-43'
            WRITE (*,60) (Tobflt(Iband),Iband=0,43)
 60         FORMAT (5F7.1)
            GOTO 80
         ENDIF
 70   CONTINUE
 80   END
      REAL FUNCTION Airab(Fr,Tc,Rh,P)
C----------------------------------------------------------------------
C     Absorption of sound in air by molecular processes.
C----------------------------------------------------------------------
C     [1]  American National Standard Method for the Calculation of the
C          Absorption of Sound by the Atmosphere, ANSI S1.26-1978.
```

```
C      [2]  H. E. Bass, L. C. Sutherland, J. Piercy and L. Evans,
C           "Absorption of Sound by the Atmosphere," in 'Physical
C           Acoustics' (1984).
C-----------------------------------------------------------------------
C      Fr -- Frequency (Hz)
C      Tc -- Temperature (deg C)
C      Rh -- Relative Humidity (%) (0 < Rh < 100)
C      P  -- Pressure (standard sea-level atm)
C-----------------------------------------------------------------------
       REAL    Alpha,Fr,Frn,Fro,H,P,Psatpo,Rh,T,T0,T20ot,Tc
C
C      -----Temperature in Kelvins, and reference temperature 20 deg C.
       T = Tc+273.15
       T0 = 293.15
       T20ot = T0/T
C
C      -----Saturation pressure ([2], Eqn. 72).
       Psatpo = 10.0**(8.422-10.05916*T20ot+5.023*ALOG10(T20ot))
C
C      -----Percent mole fraction of water vapor ([1], Eqn. D10).
       H = Rh*Psatpo/P
C
C      -----Oxygen and nitrogen relaxation frequencies ([1], Eqs. 8 & 9).
       Fro = P*(24.0+4.41E4*H*(0.05+H)/(0.391+H))
       Frn = P*SQRT(T20ot)*(9.0+350.0*H*EXP(-6.142*(T20ot**0.3333-1.0)))
C
C      -----Absorption coefficient, Alpha (Nepers/m) ([1], Eqn. 10).
       Alpha = 0.01278*EXP(-2239.1/T)/(Fro+Fr*Fr/Fro)
       Alpha = 0.10690*EXP(-3352.0/T)/(Frn+Fr*Fr/Frn) + Alpha
       Alpha = Fr*Fr*(1.84E-11/SQRT(T20ot)/P+T20ot**2.5*Alpha)
C
       Airab = Alpha
C
       RETURN
       END
       SUBROUTINE Chprof(Dtdh,W10,W10pro,Prof,S10)
C-----------------------------------------------------------------------
C      Choose the sound speed profile type from met profiles
C         - linear: Prof=0
C         - log: Prof=1
C-----------------------------------------------------------------------
       INTEGER  Prof
       REAL  Dtdh,S10,W10,W10pro
C
C      -----Decide which profile type to use.
       IF (W10.GT.2.0) THEN
C
C         -----If wind magnitude > 2 m/s use log speed profile.
          Prof = 1
          S10 = W10pro
C
       ELSE
C         -----Use a linear profile, based on the temperature.
          Prof = 0
          S10 = Dtdh*0.61*10.0
       ENDIF
C
       RETURN
       END
       REAL FUNCTION Dbsum(X,Y)
C-----------------------------------------------------------------------
C      Add energies dB(E(X)+E(Y)) of two uncorrelated signals.
C-----------------------------------------------------------------------
       REAL Lo,Hi,X,Y
C
C      -----Lower and higher values of X and Y to avoid overflow.
```

```
      Lo = AMIN1(X,Y)
      Hi = AMAX1(X,Y)
C
      IF (Lo.LE.-100.0) THEN
C        -----Don't add in values of 0.0, (zero energy flag).
         Dbsum = AMAX1(Hi,-100.0)
      ELSE
C        -----Convert to energy, add, and convert back to dB.
C        Dbsum = 10.0*ALOG10(10.0**(Hi/10.0)+10.0**(Lo/10.0))
         Dbsum = Hi+10.0*ALOG10(1.0+10.0**((Lo-Hi)/10.0))
      ENDIF
C
      RETURN
      F D
      INTEGER FUNCTION Endstr(C)
C-----------------------------------------------------------------------
C     Function returns the position of the last non-blank charater.
C-----------------------------------------------------------------------
      CHARACTER  C*80
      INTEGER  Last
C
      DO 10 Last=80,1,-1
         IF (C(Last:Last).NE.' ') GOTO 20
 10   CONTINUE
C
 20   Endstr = Last
      RETURN
      END
      SUBROUTINE Fitrh(Rhhi,Rh,Rh0)
C-----------------------------------------------------------------------
C     Find the relative humidity nearest the surface from the profile.
C-----------------------------------------------------------------------
      INTEGER  I,N
      REAL  Low,Rhhi(100),Rh(100),Rh0
C
C     -----Count up the number of data points.
      DO 10 N=0,99
         IF (Rhhi(N+1).LT.0.0) GOTO 20
 10   CONTINUE
C
 20   IF (N.EQ.0) THEN
C        -----Assume standard humidity.
         Rh0 = 70.0
      ELSE IF (N.EQ.1) THEN
C        -----Use the only available value.
         Rh0 = Rh(1)
      ELSE
C        -----Use value closest to the surface.
         Low = Rhhi(1)
         Rh0 = Rh(1)
         DO 30 I=2,N
            IF (Rhhi(I).LT.Low) THEN
               Low = Rhhi(I)
               Rh0 = Rh(I)
            ENDIF
 30      CONTINUE
      ENDIF
C
      RETURN
      END
      SUBROUTINE Fittc(Tchi,Tc,Tc0,Dtdh)
C-----------------------------------------------------------------------
C     Find the slope and intercept of the temperature profile.
C-----------------------------------------------------------------------
      REAL  Dtdh,Tc0,Tc(100),Tchi(100)
      INTEGER  N
```

```
C
C     -----Count up the number of data points.
      DO 10 N=0,99
         IF (Tchi(N+1).LT.0.0) GOTO 20
 10   CONTINUE
C
 20   IF (N.EQ.0) THEN
C        -----Assume standard temp, lapse rate.
         Dtdh = -0.0098
         Tc0 = 15.0
      ELSE IF (N.EQ.1) THEN
C        -----Assume lapse rate, find Tc at the ground.
         Dtdh = -0.0098
         Tc0 = Tc(1)-Tchi(1)*Dtdh
      ELSE
C        -----Use slope and intercept of best-fit line.
         CALL Linfit(Tchi,Tc,N,Tc0,Dtdh)
      ENDIF
C
      RETURN
      END
      SUBROUTINE Fitw(Whi,Wx,Wy,Rufhi,Azimth,W10,W10pro)
C-----------------------------------------------------------------------
C     Find the best log wind profile.
C-----------------------------------------------------------------------
      REAL   Azimth,Pi,Rufhi,Yx,Yy
      REAL   W10,W10pro,W10x,W10y,Whi(100),Whilg(100),Wx(100),Wy(100)
      INTEGER  I,N
C
C     -----Figure out Pi.
      Pi = 4.0*ATAN(1.0)
C
C     -----Count up the number of data points.
      DO 10 N=0,99
         IF (Whi(N+1).LT.0.0) GOTO 20
 10   CONTINUE
C
 20   IF (N.EQ.0) THEN
C        -----Default wind speed at 2.0 m/s along propagation azimuth.
         W10 = 2.0
         W10pro = 2.0
      ELSE
C        -----Make a log10 height array.
         DO 30 I=1,N
            Whilg(I) = ALOG10(Whi(I))
 30      CONTINUE
C
C        -----Insert false datum at the roughness height.
         Whilg(N+1) = ALOG10(Rufhi)
         Wx(N+1) = 0.0
         Wy(N+1) = 0.0
C
C        -----Least squares fit on (Whilg[],Wx[]) and (Whilg[],Wy[]).
         CALL Linfit(Whilg,Wx,N+1,Yx,W10x)
         CALL Linfit(Whilg,Wy,N+1,Yy,W10y)
C
C        -----Wind magnitude.
         W10 = SQRT(W10x**2+W10y**2)
C
C        - Wind flow along the propagation azimuth.
C        -  W10pro = (W10x + W10y) (dot) (sin(Azimth)x + cos(Azimth)y)
C
         W10pro = W10x*SIN(Azimth*Pi/180.0)+W10y*COS(Azimth*Pi/180.0)
      ENDIF
C
      RETURN
```

```
      END
      SUBROUTINE Fndfil(Tprf,Tspd,Tgrd,Tsrc,Trec,Dir,Index,Bfil)
C-------------------------------------------------------------------------
C     Find the best file for this propagation condition.
C-------------------------------------------------------------------------
C     B - Best match.
C     N - Newest match.
C     T - Target match.
C-------------------------------------------------------------------------
      CHARACTER  Dir*80,Index*80,Bfil*80,Nfil*80
      INTEGER  Endstr,Nprf,Tprf
      LOGICAL  Match
      REAL  Bspd,Nspd,Tspd,Difspd,  Bgrd,Ngrd,Tgrd,Difgrd
      REAL  Bsrc,Nsrc,Tsrc,Difsrc,  Brec,Nrec,Trec,Difrec,  Tol
      EXTERNAL Endstr
C
      Match = .FALSE.
      Tol = 0.001
C
      OPEN (10,FILE=Dir(1:Endstr(Dir))//Index,STATUS='OLD',ERR=30)
 10   READ (10,*,ERR=30,END=20) Nfil,Nprf,Nspd,Ngrd,Nsrc,Nrec
C
C     -----Skip if not target profile type.
      IF (Nprf.EQ.Tprf) THEN
C
C        -----If not yet a match, set these best parameters.
         IF (.NOT.Match) THEN
            Bspd = Nspd
            Bgrd = Ngrd
            Bsrc = Nsrc
            Brec = Nrec
            Bfil = Nfil
            Match = .TRUE.
         ELSE
C
C           -----There is at least one match; fit best speed.
            Difspd = ABS(Nspd-Tspd)-ABS(Bspd-Tspd)
            IF (Difspd.LT.0.0) THEN
               Bspd = Nspd
               Bgrd = Ngrd
               Bsrc = Nsrc
               Brec = Nrec
               Bfil = Nfil
            ELSE IF (Difspd.LT.Tol) THEN
C
C              -----Same speed; fit best ground type.
               Difgrd = ABS(Ngrd-Tgrd)-ABS(Bgrd-Tgrd)
               IF (Difgrd.LT.0.0) THEN
                  Bgrd = Ngrd
                  Bsrc = Nsrc
                  Brec = Nrec
                  Bfil = Nfil
               ELSE IF (Difgrd.LT.Tol) THEN
C
C                 -----Same speed,ground; fit best source height.
                  Difsrc = ABS(Nsrc-Tsrc)-ABS(Bsrc-Tsrc)
                  IF (Difsrc.LT.0.0) THEN
                     Bsrc = Nsrc
                     Brec = Nrec
                     Bfil = Nfil
                  ELSE IF (Difsrc.LT.Tol) THEN
C
C                    -----Same speed, ground, source; fit best rec.
                     Difrec = ABS(Nrec-Trec)-ABS(Brec-Trec)
                     IF (Difrec.LT.0.0) THEN
                        Brec = Nrec
```

```
                        Bfil = Nfil
                     ENDIF
                  ENDIF
               ENDIF
            ENDIF
         ENDIF
      ENDIF
      GOTO 10
C
C     -----Flag a missing decay table as a fatal error.
 20   IF (.NOT.Match) THEN
         PRINT *,'ERROR: No decay table match found.'
         WRITE (*,*) 'INDEX FILE NAME:',Dir(1:Endstr(Dir))//Index
         STOP
      ENDIF
      RETURN
C
C     -----Some kind of reading error.
 30   WRITE (*,*) 'ERROR: bad entry in index file.'
      WRITE (*,*) 'INDEX FILE NAME:',Dir(1:Endstr(Dir))//Index
      WRITE (*,*) 'expecting file name with parameters,'
      WRITE (*,*) 'instead found:'
      WRITE (*,*) 'FILE NAME:',Nfil
      WRITE (*,*) 'PROFILE TYPE:',Nprf
      WRITE (*,*) 'PROFILE SPEED:',Nspd
      WRITE (*,*) 'GROUND CLASS:',Ngrd
      WRITE (*,*) 'SOURCE HEIGHT:',Nsrc
      WRITE (*,*) 'RECEIVER HEIGHT:',Nrec
      STOP
      END
      SUBROUTINE Frwtng(TobIn,TobAwt,TobCwt,TobUwt,Usrcrv)
C-----------------------------------------------------------------------
C     Apply frequency weighting curve to 1/3-octave bands 0 to 43.
C-----------------------------------------------------------------------
      INTEGER  Iband
      REAL  Acrv(0:43),Ccrv(0:43),Usrcrv(0:43)
      REAL  TobIn(0:43),TobAwt(0:43),TobCwt(0:43),TobUwt(0:43)
C
      DATA  Acrv/
     >  -148.5,-140.5,-132.6,-124.6,-116.6,
     >  -108.7,-100.8, -93.0, -85.3, -77.7,
     >   -70.4, -63.3, -56.6, -50.4, -44.7,
     >   -39.4, -34.6, -30.2, -26.1, -22.5,
     >   -19.1, -16.0, -13.3, -10.8,  -8.6,
     >    -6.6,  -4.8,  -3.2,  -1.9,  -0.8,
     >     0.0,   0.5,   0.9,   1.1,   1.2,
     >     1.1,   0.9,   0.5,  -0.1,  -1.1,
     >    -2.4,  -4.3,  -6.6,  -9.3/
C
      DATA  Ccrv/
     >     -52.5, -48.5, -44.5, -40.5, -36.6,
     >     -32.6, -28.8, -24.9, -21.2, -17.6,
     >     -14.3, -11.2,  -8.5,  -6.2,  -4.4,
     >      -3.0,  -1.9,  -1.2,  -0.8,  -0.5,
     >      -0.2,  -0.1,   0.0,   0.0,   0.0,
     >       0.0,   0.0,   0.0,   0.0,   0.0,
     >       0.0,   0.0,   0.0,  -0.1,  -0.2,
     >      -0.5,  -0.8,  -1.2,  -1.9,  -3.0,
     >      -4.4,  -6.2,  -8.5, -11.2/
C
C     -----Apply weightings to the spectra.
      DO 10 Iband=0,43
         TobAwt(Iband) = AMAX1(-100.0,TobIn(Iband)+Acrv(Iband))
         TobCwt(Iband) = AMAX1(-100.0,TobIn(Iband)+Ccrv(Iband))
         TobUwt(Iband) = AMAX1(-100.0,TobIn(Iband)+Usrcrv(Iband))
 10   CONTINUE
```

```
C
      RETURN
      END
      SUBROUTINE Hold(X,Y)
C-----------------------------------------------------------------------
C     Copy one spectrum X into spectrum Y.
C-----------------------------------------------------------------------
      INTEGER  Iband
      REAL  X(0:43),Y(0:43)
C
      DO 10 Iband=0,43
         Y(Iband) = X(Iband)
 10   CONTINUE
C
      RETURN
      END
      SUBROUTINE Linfit(X,Y,Ndata,Y0,Slope)
C-----------------------------------------------------------------------
C     Least squares linear fit y=Slope*x+Y0 to data y(i),x(i),i=1,Ndata.
C-----------------------------------------------------------------------
C     [1]  Press, et al., Numerical Recipes, Cambridge, pp. 504-509.
C-----------------------------------------------------------------------
      INTEGER  I,Ndata
      REAL  X(Ndata),Y(Ndata),Y0,Slope, Sxon,Sx,Sy,St2,T
C
      Sx = 0.0
      Sy = 0.0
      St2 = 0.0
      Slope = 0.0
C
C     -----Sum over X and over Y.
      DO 10 I=1,Ndata
         Sx = Sx+X(I)
         Sy = Sy+Y(I)
 10   CONTINUE
      Sxon = Sx/FLOAT(Ndata)
C
C     -----Sum of squared differences.
      DO 20 I=1,Ndata
         T = X(I)-Sxon
         St2 = St2+T*T
         Slope = Slope+T*Y(I)
 20   CONTINUE
C
C     -----Intercept and slope.
      Slope = Slope/St2
      Y0 = (Sy-Sx*Slope)/FLOAT(Ndata)
C
      RETURN
      END
      SUBROUTINE Ntrpl8(A,X,B,Y,C,Z)
C-----------------------------------------------------------------------
C     Interpolates between spectra X and Y, based on value of C in (A,B).
C-----------------------------------------------------------------------
      INTEGER  Iband
      REAL  Ratio,A,B,C,X(0:43),Y(0:43),Z(0:43)
C
C     -----Find the fraction of C of the way between A and B.
      Ratio = (C-A)/(B-A)
C
C     -----Let's just don't extrapolate.
      IF (Ratio.GT.1.0.OR.Ratio.LT.0.0) THEN
         WRITE (*,*) 'ERROR: interpolation attempted outside interval'
         WRITE (*,*) 'looking for:',C,'in interval:',CMPLX(A,B)
         STOP
      ENDIF
```

```
C
C     -----Do all of the bands.
      DO 10 Iband=0,43
         Z(Iband) = X(Iband) + (Y(Iband)-X(Iband))*Ratio
 10   CONTINUE
C
      RETURN
      END
      REAL FUNCTION Ovrall(Tob)
C-----------------------------------------------------------------------
C     Sum over the 1/3-octave bands 0 to 43.
C-----------------------------------------------------------------------
      INTEGER  Iband
      REAL  Tob(0:43), Leq
      REAL  Dbsum
      EXTERNAL Dbsum
C
C     -----Find the overall level.
      Leq = -100.0
      DO 10 Iband=1,43
         Leq = Dbsum(Tob(Iband),Leq)
 10   CONTINUE
C
      Ovrall = Leq
C
      RETURN
      END
      SUBROUTINE Patch(X,Bad)
C-----------------------------------------------------------------------
C     Fill outside band edge and interpolate to fill holes in spectrum.
C-----------------------------------------------------------------------
      INTEGER  Iband,Jband,First,Last
      LOGICAL  Bad
      REAL  X(0:43),Ratio
      Bad = .FALSE.
C
C     -----Find the first good band.
      DO 10 First=0,43
         IF (X(First).LT.100.0) GOTO 20
 10   CONTINUE
C
C     -----All bands are bad.
      Bad = .TRUE.
      RETURN
C
C     -----Find the last good band.
 20   DO 30 Last=43,0,-1
         IF (X(Last).LT.100.0) GOTO 40
 30   CONTINUE
C
C     -----Fill all bands on the outer edges.
 40   DO 50 Iband=First-1,0,-1
         X(Iband) = X(First)
 50   CONTINUE
C
      DO 60 Iband=Last+1,43
         X(Iband) = X(Last)
 60   CONTINUE
C
C     -----Scan for holes in the spectrum.
      DO 90 Iband=First+1,Last-1
C
C        -----Check if missing point.
         IF (X(Iband).GE.100.0) THEN
C
C           -----Find the next good band (last band is OK).
```

```
      DO 70 Jband=Iband+1,Last
         IF (X(Jband).LT.100.0) GOTO 80
 70   CONTINUE
C
C     -----Use the Jband and Iband-1 to interpolate on.
 80   Ratio = 1.0/FLOAT(Jband-Iband+1)
      X(Iband) = X(Iband-1)+(X(Jband)-X(Iband-1))*Ratio
      ENDIF
C
 90   CONTINUE
C
      RETURN
      END
      SUBROUTINE Rdexs(Rnum,Exsdb,Fopen,Fend,Dir,Best)
C---------------------------------------------------------------------
C     Read the excess levels file.
C---------------------------------------------------------------------
      CHARACTER  Best*80,Dir*80,Header*20
      INTEGER  Iband,Endstr,Rnum
      LOGICAL  Fend,Fopen,Bad
      REAL  Exsdb(0:43)
      EXTERNAL Endstr
C
C     -----Take care of opening the "Best" file.
      IF (.NOT.Fopen) THEN
         OPEN (20,File=Dir(1:Endstr(Dir))//Best,STATUS='OLD',ERR=10)
         Fend = .FALSE.
         Fopen = .TRUE.
         REWIND (UNIT=20)
      ENDIF
C
C     -----Read the range.
 1    READ (20,5,ERR=20,END=40) Header
 5    FORMAT (A20)
      READ (20,*) Rnum
C
C     -----Read average spectrum, bands 0-4, 5-9, etc. through 43.
      READ (20,*,END=20)
      READ (20,*,ERR=20,END=20) (Exsdb(Iband),Iband=0,43)
C
C     -----Try and patch empty holes in the decay spectra.
      CALL Patch(Exsdb,Bad)
      IF (Bad) GOTO 1
      RETURN
C
C     -----Bad file name or missing file error.
 10   WRITE (*,*) 'ERROR: decay file doesn''t open:'
      WRITE (*,*) 'Directory/File:',Dir(1:Endstr(Dir))//Best
      STOP
C
C     -----We found end of file too early, or bad numbers.
 20   WRITE (*,*) 'ERROR: early end of file or bad spectrum:'
      WRITE (*,*) 'Directory/File:',Dir(1:Endstr(Dir))//Best
      WRITE (*,*) 'Range number:',Rnum
      WRITE (*,*) 'Spectrum:'
      WRITE (*,30) (Exsdb(Iband),Iband=0,43)
 30   FORMAT (5F7.1)
      STOP
C
C     -----Found end of file.
 40   Fend = .TRUE.
      RETURN
      END
      SUBROUTINE Rdgeom(Srchi,Rechi,Range,Azimth)
C---------------------------------------------------------------------
C     Read in the geometry of the propagation problem.
```

```
C-----------------------------------------------------------------------
      CHARACTER  Header*20
      REAL  Azimth,Range,Srchi,Rechi
 10   FORMAT (A20)
C
C     -----Source height (m).
      READ (*,10,ERR=30,END=20) Header
      IF (Header(1:13).NE.'SOURCE HEIGHT') GOTO 30
      READ (*,*,ERR=30,END=20) Srchi
C
C     -----Reciever height (m).
      READ (*,10,ERR=30,END=20) Header
      IF (Header(1:15).NE.'RECEIVER HEIGHT') GOTO 30
      READ (*,*,ERR=30,END=20) Rechi
      IF (Srchi.LT.0.0.OR.Rechi.LT.0.0) THEN
         WRITE (*,*) 'ERROR: Source and receiver must be above ground.'
         WRITE (*,*) 'Instead, found:',Srchi,Rechi
         STOP
      ENDIF
C
C     -----Range for the prediction (m)  (0=ALL).
      READ (*,10,ERR=30,END=20) Header
      IF (Header(1:14).NE.'RECEIVER RANGE') GOTO 30
      READ (*,*,ERR=30,END=20) Range
C
C     -----Azimuth for propagation (deg).
      READ (*,10,ERR=30,END=20) Header
      IF (Header(1:16).NE.'RECEIVER AZIMUTH') GOTO 30
      READ (*,*,ERR=30,END=20) Azimth
      RETURN
C
 20   Header = 'EOF'
 30   WRITE (*,40) Srchi,Rechi,Range,Azimth,Header
 40   FORMAT('ERROR: expecting a header, followed by a value:'/
     >   5X,'SOURCE HEIGHT'/  F15.2/  5X,'RECEIVER HEIGHT'/  F15.2/
     >   5X,'RECEIVER RANGE'/ F15.2/  5X,'RECEIVER AZIMUTH'/ F15.2/
     >   'error found with:',A20)
      STOP
      END
      SUBROUTINE Rdgrnd(Grclas,Rufhi)
C-----------------------------------------------------------------------
C     Read in the ground conditions.
C-----------------------------------------------------------------------
      CHARACTER  Header*20
      REAL  Grclas,Rufhi
 10   FORMAT (A20)
C
C     -----Ground classification type (hard=0, soft=1).
      READ (*,10,ERR=30,END=20) Header
      IF (Header(1:12).NE.'GROUND CLASS') GOTO 30
      READ (*,*,ERR=30,END=20) Grclas
      IF (Grclas.LT.0.0.OR.Grclas.GT.1.0) THEN
         WRITE (*,*) 'ERROR: Ground class must be between 0.0 and 1.0.'
         WRITE (*,*) 'Instead, found:',Grclas
         STOP
      ENDIF
C
C     -----Ground roughness height (m).
      READ (*,10,ERR=30,END=20) Header
      IF (Header(1:12).NE.'GROUND ROUGH') GOTO 30
      READ (*,*,ERR=30,END=20) Rufhi
      IF (Rufhi.LE.0.0) THEN
         WRITE (*,*) 'ERROR: Ground roughness height must exceed 0.0.'
         WRITE (*,*) 'Instead found:',Rufhi
         STOP
      ENDIF
```

```
      RETURN
C
 20   Header = 'EOF'
 30   WRITE (*,40) Grclas,Rufhi,Header
 40   FORMAT('ERROR: expecting a header, followed by a value:'/
     >   5X,'GROUND CLASS'/   F15.2/  5X,'GROUND ROUGH'/     F15.2/
     >   'error found with:',A20)
      STOP
      END
      SUBROUTINE Rdlvls(TobSrc,Refdis,Usrcrv)
C-----------------------------------------------------------------------
C     Calls routines to read source spectrum and weighting.
C-----------------------------------------------------------------------
      CHARACTER  Header*20
      INTEGER  Iband
      REAL  Refdis,TobSrc(0:43),Usrcrv(0:43)
C
C     ------Source spectrum.
      READ (*,10,ERR=30,END=20) Header
 10   FORMAT (A20)
      IF (Header(1:18).NE.'SOURCE LEVEL SPECT') GOTO 30
      READ (*,*,ERR=30,END=20) (TobSrc(Iband),Iband=0,43)
C
C     ------Reference distance.
      READ (*,10,ERR=60,END=50) Header
      IF (Header(1:14).NE.'REFERENCE DIST') GOTO 60
      READ (*,*,ERR=60,END=50) Refdis
      IF (Refdis.LE.0.0) GOTO 60
C
C     ------User weighting spectrum.
      READ (*,10,ERR=80,END=70) Header
      IF (Header(1:20).NE.'USER WEIGHTING SPECT') GOTO 80
      READ (*,*,ERR=80,END=70) (Usrcrv(Iband),Iband=0,43)
      RETURN
C
C     -----Error in source spectrum.
 20   Header = 'EOF'
 30   WRITE (*,*) 'ERROR: expecting source spectrum.'
      WRITE (*,*) 'Instead, found:',Header
      WRITE (*,*) 'Spectrum ='
      WRITE (*,40) (TobSrc(Iband),Iband=0,43)
 40   FORMAT (5F7.1)
      STOP
C
C     -----Error in reference distance.
 50   Header = 'EOF'
 60   WRITE (*,*) 'ERROR: expecting positive reference distance.'
      WRITE (*,*) 'Instead, found:',Header,' Refdis=',Refdis
      STOP
C
C     -----Error in user weighting spectrum.
 70   Header = 'EOF'
 80   WRITE (*,*) 'ERROR: expecting user weighting spectrum.'
      WRITE (*,*) 'Instead, found:',Header
      WRITE (*,*) 'Spectrum ='
      WRITE (*,40) (Usrcrv(Iband),Iband=0,43)
      STOP
C
      END
      SUBROUTINE Rdname(Dir,Index)
C-----------------------------------------------------------------------
C     Reads the directory name and index file name from standard input.
C-----------------------------------------------------------------------
      CHARACTER  Dir*80,Header*20,Index*80
C
C     -----Read the directory name header.
```

```
      READ (*,5,ERR=30,END=20) Header
 5    FORMAT (A20)
 10   FORMAT (A80)
      IF (Header(1:6).NE.'LOOKUP') GOTO 30
      READ (*,10,ERR=30,END=20) Dir
C
C     -----Read the index file header.
      READ (*,5,ERR=30,END=20) Header
      IF (Header(1:17).NE.'LOOKUP INDEX FILE') GOTO 30
      READ (*,10,ERR=30,END=20) Index
      RETURN
C
C     -----Report some error and bail out.
 20   Header = 'EOF'
 30   WRITE (*,40) Dir,Index,Header
 40   FORMAT('ERROR: expecting a header, followed by a string:'/
     >   5x,'DIRECTORY NAME'/   A60/  5x,'INDEX FILE NAME'/   A60/
     >   5x,'with:',A20)
      STOP
      END
      SUBROUTINE Rdrh(Rhhi,Rh)
C-----------------------------------------------------------------------
C     Read the relative humidity measurements into an array.
C-----------------------------------------------------------------------
      CHARACTER  Header*20
      INTEGER  N
      REAL  Rhhi(100),Rh(100)
C
C     -----Skip rh title line.
      READ (*,10,ERR=50,END=40) Header
 10   FORMAT (A20)
      IF (Header(1:16).NE.'HUMIDITY PROFILE') GOTO 50
C
C     -----Read one humidity datum.
      N = 1
 20   READ (*,*,ERR=30,END=40) Rhhi(N),Rh(N)
C
C     -----Negative height flag means end of the list; returns.
      IF (Rhhi(N).LT.0.0) RETURN
C
C     -----Check that the humidity is between 1 and 99 percent.
      IF (Rh(N).LE.1.0.OR.Rh(N).GT.99.9) GOTO 30
C
C     -----Ready to fill next element, then over-write last until neg.
      IF (N.LT.100) N = N+1
      GOTO 20
C
C     -----Error for unreasonable humidity.
 30   WRITE (*,*) 'ERROR: Rh must be in the range 1 to 99 percent.'
      WRITE (*,*) 'Instead, found:',Rh(N),' N=',N
      STOP
C
C     -----Other kinds of missing or bad input for humidity.
 40   Header = 'EOF'
 50   WRITE (*,*) 'ERROR: expected humidity profile.'
      WRITE (*,*) 'Instead, found:',Header
      STOP
      END
      SUBROUTINE Rdtc(Tchi,Tc)
C-----------------------------------------------------------------------
C     Read the temperature measurements into an array.
C-----------------------------------------------------------------------
      CHARACTER  Header*20
      INTEGER  N
      REAL  Tchi(100),Tc(100)
C
```

```
C     -----Temperature profile title line.
      READ (*,10,ERR=50,END=40) Header
 10   FORMAT (A20)
      IF (Header(1:19).NE.'TEMPERATURE PROFILE') GOTO 50
C
C     -----Read one temperature datum.
      N = 1
 20   READ (*,*,ERR=30,END=40) Tchi(N),Tc(N)
C
C     -----Negative height means end of the list.
      IF (Tchi(N).LT.0.0) RETURN
C
C     -----Check that the temperature is reasonable.
      IF (Tc(N).LT.-50.0.OR.Tc(N).GT.50.0) GOTO 30
C
C     -----Ready to fill next element.  Over-write #100 until neg. height.
      IF (N.LT.100) N = N+1
      GOTO 20
C
C     -----Error for unreasonable temperature.
 30   WRITE (*,*) 'ERROR: Tc must be in the range -50 C to 50 C.'
      WRITE (*,*) 'Instead, found:',Tc(N),' N=',N
      STOP
C
C     -----Other kinds of missing or bad input for temperature.
 40   Header = 'EOF'
 50   WRITE (*,*) 'ERROR: expected temperature profile.'
      WRITE (*,*) 'Instead, found:',Header
      STOP
      END
      SUBROUTINE Rdtob(Tob)
C-----------------------------------------------------------------------
C     Read a one-third octave band spectrum, bands 0 - 43.
C-----------------------------------------------------------------------
      INTEGER  I
      REAL  Tob(0:43)
C
C     -----Read the spectrum, bands 0 - 9, 10 - 19, etc.
      READ (*,*,ERR=100) (Tob(I),I=0,43)
      RETURN
C
 100  WRITE (*,*) 'ERROR: bad number in spectrum'
      STOP
      END
      SUBROUTINE Rdver
C-----------------------------------------------------------------------
C     Read the version number for Lookup.
C-----------------------------------------------------------------------
      CHARACTER  Header*20
      REAL  Ver
C
C     -----Read the title line and check if we've got the right input.
      READ (*,10,ERR=40,END=30) Header
 10   FORMAT (A20)
      IF (Header(1:19).NE.'LOOKUP LEVELS INPUT') GOTO 40
C
C     -----Check the version number.
      Ver = 0.0
      READ (*,*,ERR=20,END=30) Ver
C
C     -----Return if we have the right input file.
      IF (Ver.EQ.1.0) RETURN
C
C     -----Error for bad file version.
 20   WRITE (*,*) 'ERROR: expecting Lookup program version: 1.0.'
      WRITE (*,*) 'Instead, found:',Ver
```

```
      STOP
C
C     -----Error for other bad or missing input.
 30   Header = 'EOF'
 40   WRITE (*,*) 'ERROR: expecting Lookup input header.'
      WRITE (*,*) 'Instead, found:',Header
      STOP
      END
      SUBROUTINE Rdwswd(Whi,Wx,Wy)
C-----------------------------------------------------------------------
C     Read the wind speed and direction measurements into arrays.
C-----------------------------------------------------------------------
      CHARACTER  Header*20
      INTEGER  N
      REAL  Pi,Whi(100),Wdir(100),Ws(100),Wx(100),Wy(100)
C
C     -----Compute Pi.
      Pi = 4.0*ATAN(1.0)
C
C     -----Skip a the wind profile title line.
      READ (*,10,ERR=50,END=40) Header
 10   FORMAT (A20)
      IF (Header(1:12).NE.'WIND PROFILE') GOTO 50
C
C     -----Read one wind speed, wind direction datum.
      N = 1
 20   READ (*,*,ERR=30,END=40) Whi(N),Ws(N),Wdir(N)
C
C     -----Negative height means end of the list; returns.
      IF (Whi(N).LT.0.0) RETURN
C
C     -----Don't allow negative wind speeds or speeds above 40 m/s.
      IF (Ws(N).LT.0.0.OR.Ws(N).GT.40.0) GOTO 30
C
C     -----Use elements 1 to 99, only.
      IF (N.LT.100) THEN
C
C        -----Convert to Wx, and Wy. (Wdir is meteo. wind direction!!)
         Wx(N)  = -Ws(N)*SIN(Wdir(N)/180.0*Pi)
         Wy(N)  = -Ws(N)*COS(Wdir(N)/180.0*Pi)
C
C        -----Ready to fill next element.
         N = N+1
      ENDIF
      GOTO 20
C
C     ----Error for unreasonable wind speed or illegal numbers.
 30   WRITE (*,*) 'ERROR: Ws must be in the range 0 to 40 m/s'
      WRITE (*,*) 'Instead, found: Ws(N)=',Ws(N),' N=',N
      STOP
C
C     -----Other kinds of missing or bad input for wind.
 40   Header = 'EOF'
 50   WRITE (*,*) 'ERROR: expected wind profile.'
      WRITE (*,*) 'Instead, found:',Header
      STOP
      END
```

APPENDIX B: Sample Input File

```
LOOKUP LEVELS INPUT FILE, VERSION:
1.0
LOOKUP DIRECTORY NAME:
/usr/white/Prog/Lookup/
LOOKUP INDEX FILE:
index10.fil
SOURCE HEIGHT (m)
11.7
RECEIVER HEIGHT (m)
2.1
RECEIVER RANGE (m) (0=ALL)
0.0
RECEIVER AZIMUTH (deg)
90.0
GROUND CLASSIFICATION (HARD=0, POROUS=1)
0.51
GROUND ROUGHNESS HEIGHT (m)
0.1
TEMPERATURE PROFILE; HEIGHT(m) TEMPERATURE (deg C)
0.1  9.2
1.0  10.0
1.4  10.4
20.0 5.3
100.0   3.3
200.0   -1.0
-1.0 -1.0  THE END
HUMIDITY PROFILE; HEIGHT (m) REL. HUMIDITY (%)
0.5  50.0
1.0  55.0
10.0 45.0
100.0   55.0
200.0   20.0
-1.0 -1.0  THE END
WIND PROFILE; HEIGHT (m), WIND SPEED (m/s), DIRECTION (deg)
2.0  4.2   240.0
5.0  6.0   300.0
-1.0 -1.0  -1.0 THE END
SOURCE LEVEL SPECTRUM (dB); 1/3 OCTAVE BANDS 0-43
0.0  0.0   0.0  0.0   0.0
70.0 73.0  76.0 79.0  82.0
85.0 88.0  91.0 94.0  97.0
100.0   97.0 94.0 91.0  88.0
85.0 82.0  79.0 76.0  73.0
70.0 0.0   0.0  0.0   0.0
0.0  0.0   0.0  0.0   0.0
0.0  0.0   0.0  0.0   0.0
0.0  0.0   0.0  0.0
REFERENCE DIST (m)
250.0
USER WEIGHTING SPECTRUM (dB); 1/3 OCTAVE BANDS 0-43
0.0  0.0   0.0  0.0   0.0
0.0  0.0   0.0  0.0   0.0
10.0 10.0  10.0 10.0  10.0
0.0  0.0   0.0  0.0   0.0
0.0  0.0   0.0  0.0   0.0
0.0  0.0   0.0  0.0   0.0
0.0  0.0   0.0  0.0   0.0
0.0  0.0   0.0  0.0   0.0
0.0  0.0   0.0  0.0
```

```
LOOKUP LEVELS INPUT FILE, VERSION:
1.0
LOOKUP DIRECTORY NAME:
/usr/white/Prog/Lookup/
LOOKUP INDEX FILE:
index10.fil
SOURCE HEIGHT (m)
11.7
RECEIVER HEIGHT (m)
2.1
RECEIVER RANGE (m) (0=ALL)
10.0
RECEIVER AZIMUTH (deg)
90.0
GROUND CLASSIFICATION (HARD=0, POROUS=1)
0.51
GROUND ROUGHNESS HEIGHT (m)
0.1
TEMPERATURE PROFILE; HEIGHT(m) TEMPERATURE (deg C)
0.1  9.2
1.0  10.0
1.4  10.4
20.0 5.3
100.0   3.3
200.0   -1.0
-1.0 -1.0  THE END
HUMIDITY PROFILE; HEIGHT (m) REL. HUMIDITY (%)
0.5  50.0
1.0  55.0
10.0 45.0
100.0   55.0
200.0   20.0
-1.0 -1.0  THE END
WIND PROFILE; HEIGHT (m), WIND SPEED (m/s), DIRECTION (deg)
2.0  4.2   240.0
5.0  6.0   300.0
-1.0 -1.0  -1.0 THE END
SOURCE LEVEL SPECTRUM (dB); 1/3 OCTAVE BANDS 0-43
0.0  0.0   0.0  0.0  0.0
70.0 73.0  76.0 79.0 82.0
85.0 88.0  91.0 94.0 97.0
100.0   97.0 94.0 91.0 88.0
85.0 82.0  79.0 76.0 73.0
70.0 0.0   0.0  0.0  0.0
0.0  0.0   0.0  0.0  0.0
0.0  0.0   0.0  0.0  0.0
0.0  0.0   0.0  0.0
REFERENCE DIST (m)
250.0
USER WEIGHTING SPECTRUM (dB); 1/3 OCTAVE BANDS 0-43
0.0  0.0   0.0  0.0  0.0
0.0  0.0   0.0  0.0  0.0
10.0 10.0  10.0 10.0 10.0
0.0  0.0   0.0  0.0  0.0
0.0  0.0   0.0  0.0  0.0
0.0  0.0   0.0  0.0  0.0
0.0  0.0   0.0  0.0  0.0
0.0  0.0   0.0  0.0  0.0
0.0  0.0   0.0  0.0
```

APPENDIX C: Sample Index File

'ahnn14'	0	-14	0	2	1.2
'bhnn14'	0	-14	0	5	1.2
'chnn14'	0	-14	0	10	1.2
'dhnn14'	0	-14	0	100	1.2
'asnn14'	0	-14	1	2	1.2
'bsnn14'	0	-14	1	5	1.2
'csnn14'	0	-14	1	10	1.2
'dsnn14'	0	-14	1	100	1.2
'ahnn12'	0	-12	0	2	1.2
'bhnn12'	0	-12	0	5	1.2
'chnn12'	0	-12	0	10	1.2
'dhnn12'	0	-12	0	100	1.2
'asnn12'	0	-12	1	2	1.2
'bsnn12'	0	-12	1	5	1.2
'csnn12'	0	-12	1	10	1.2
'dsnn12'	0	-12	1	100	1.2
'ahnn10'	0	-10	0	2	1.2
'bhnn10'	0	-10	0	5	1.2
'chnn10'	0	-10	0	10	1.2
'dhnn10'	0	-10	0	100	1.2
'asnn10'	0	-10	1	2	1.2
'bsnn10'	0	-10	1	5	1.2
'csnn10'	0	-10	1	10	1.2
'dsnn10'	0	-10	1	100	1.2
'ahnn08'	0	-8	0	2	1.2
'bhnn08'	0	-8	0	5	1.2
'chnn08'	0	-8	0	10	1.2
'dhnn08'	0	-8	0	100	1.2
'asnn08'	0	-8	1	2	1.2
'bsnn08'	0	-8	1	5	1.2
'csnn08'	0	-8	1	10	1.2
'dsnn08'	0	-8	1	100	1.2
'ahnn06'	0	-6	0	2	1.2
'bhnn06'	0	-6	0	5	1.2
'chnn06'	0	-6	0	10	1.2
'dhnn06'	0	-6	0	100	1.2
'asnn06'	0	-6	1	2	1.2
'bsnn06'	0	-6	1	5	1.2
'csnn06'	0	-6	1	10	1.2
'dsnn06'	0	-6	1	100	1.2
'ahnn05'	0	-5	0	2	1.2
'bhnn05'	0	-5	0	5	1.2
'chnn05'	0	-5	0	10	1.2
'dhnn05'	0	-5	0	100	1.2
'asnn05'	0	-5	1	2	1.2
'bsnn05'	0	-5	1	5	1.2
'csnn05'	0	-5	1	10	1.2
'dsnn05'	0	-5	1	100	1.2
'ahnn04'	0	-4	0	2	1.2
'bhnn04'	0	-4	0	5	1.2
'chnn04'	0	-4	0	10	1.2
'dhnn04'	0	-4	0	100	1.2
'asnn04'	0	-4	1	2	1.2
'bsnn04'	0	-4	1	5	1.2
'csnn04'	0	-4	1	10	1.2
'dsnn04'	0	-4	1	100	1.2
'ahnn02'	0	-2	0	2	1.2
'bhnn02'	0	-2	0	5	1.2
'chnn02'	0	-2	0	10	1.2
'dhnn02'	0	-2	0	100	1.2
'asnn02'	0	-2	1	2	1.2
'bsnn02'	0	-2	1	5	1.2

'csnn02'	0	-2	1	10	1.2
'dsnn02'	0	-2	1	100	1.2
'ahnn01'	0	-1	0	2	1.2
'bhnn01'	0	-1	0	5	1.2
'chnn01'	0	-1	0	10	1.2
'dhnn01'	0	-1	0	100	1.2
'asnn01'	0	-1	1	2	1.2
'bsnn01'	0	-1	1	5	1.2
'csnn01'	0	-1	1	10	1.2
'dsnn01'	0	-1	1	100	1.2
'ahnp00'	0	0	0	2	1.2
'bhnp00'	0	0	0	5	1.2
'chnp00'	0	0	0	10	1.2
'dhnp00'	0	0	0	100	1.2
'asnp00'	0	0	1	2	1.2
'bsnp00'	0	0	1	5	1.2
'csnp00'	0	0	1	10	1.2
'dsnp00'	0	0	1	100	1.2
'ahnp01'	0	1	0	2	1.2
'bhnp01'	0	1	0	5	1.2
'chnp01'	0	1	0	10	1.2
'dhnp01'	0	1	0	100	1.2
'asnp01'	0	1	1	2	1.2
'bsnp01'	0	1	1	5	1.2
'csnp01'	0	1	1	10	1.2
'dsnp01'	0	1	1	100	1.2
'ahnp02'	0	2	0	2	1.2
'bhnp02'	0	2	0	5	1.2
'chnp02'	0	2	0	10	1.2
'dhnp02'	0	2	0	100	1.2
'asnp02'	0	2	1	2	1.2
'bsnp02'	0	2	1	5	1.2
'csnp02'	0	2	1	10	1.2
'dsnp02'	0	2	1	100	1.2
'ahnp04'	0	4	0	2	1.2
'bhnp04'	0	4	0	5	1.2
'chnp04'	0	4	0	10	1.2
'dhnp04'	0	4	0	100	1.2
'asnp04'	0	4	1	2	1.2
'bsnp04'	0	4	1	5	1.2
'csnp04'	0	4	1	10	1.2
'dsnp04'	0	4	1	100	1.2
'ahnp05'	0	5	0	2	1.2
'bhnp05'	0	5	0	5	1.2
'chnp05'	0	5	0	10	1.2
'dhnp05'	0	5	0	100	1.2
'asnp05'	0	5	1	2	1.2
'bsnp05'	0	5	1	5	1.2
'csnp05'	0	5	1	10	1.2
'dsnp05'	0	5	1	100	1.2
'ahnp06'	0	6	0	2	1.2
'bhnp06'	0	6	0	5	1.2
'chnp06'	0	6	0	10	1.2
'dhnp06'	0	6	0	100	1.2
'asnp06'	0	6	1	2	1.2
'bsnp06'	0	6	1	5	1.2
'csnp06'	0	6	1	10	1.2
'dsnp06'	0	6	1	100	1.2
'ahnp08'	0	8	0	2	1.2
'bhnp08'	0	8	0	5	1.2
'chnp08'	0	8	0	10	1.2
'dhnp08'	0	8	0	100	1.2
'asnp08'	0	8	1	2	1.2
'bsnp08'	0	8	1	5	1.2
'csnp08'	0	8	1	10	1.2
'dsnp08'	0	8	1	100	1.2

'ahnp10'	0	10	0	2	1.2
'bhnp10'	0	10	0	5	1.2
'chnp10'	0	10	0	10	1.2
'dhnp10'	0	10	0	100	1.2
'asnp10'	0	10	1	2	1.2
'bsnp10'	0	10	1	5	1.2
'csnp10'	0	10	1	10	1.2
'dsnp10'	0	10	1	100	1.2
'ahnp12'	0	12	0	2	1.2
'bhnp12'	0	12	0	5	1.2
'chnp12'	0	12	0	10	1.2
'dhnp12'	0	12	0	100	1.2
'asnp12'	0	12	1	2	1.2
'bsnp12'	0	12	1	5	1.2
'csnp12'	0	12	1	10	1.2
'dsnp12'	0	12	1	100	1.2
'ahnp14'	0	14	0	2	1.2
'bhnp14'	0	14	0	5	1.2
'chnp14'	0	14	0	10	1.2
'dhnp14'	0	14	0	100	1.2
'asnp14'	0	14	1	2	1.2
'bsnp14'	0	14	1	5	1.2
'csnp14'	0	14	1	10	1.2
'dsnp14'	0	14	1	100	1.2
'ahgn14'	1	-14	0	2	1.2
'bhgn14'	1	-14	0	5	1.2
'chgn14'	1	-14	0	10	1.2
'dhgn14'	1	-14	0	100	1.2
'asgn14'	1	-14	1	2	1.2
'bsgn14'	1	-14	1	5	1.2
'csgn14'	1	-14	1	10	1.2
'dsgn14'	1	-14	1	100	1.2
'ahgn12'	1	-12	0	2	1.2
'bhgn12'	1	-12	0	5	1.2
'chgn12'	1	-12	0	10	1.2
'dhgn12'	1	-12	0	100	1.2
'asgn12'	1	-12	1	2	1.2
'bsgn12'	1	-12	1	5	1.2
'csgn12'	1	-12	1	10	1.2
'dsgn12'	1	-12	1	100	1.2
'ahgn10'	1	-10	0	2	1.2
'bhgn10'	1	-10	0	5	1.2
'chgn10'	1	-10	0	10	1.2
'dhgn10'	1	-10	0	100	1.2
'asgn10'	1	-10	1	2	1.2
'bsgn10'	1	-10	1	5	1.2
'csgn10'	1	-10	1	10	1.2
'dsgn10'	1	-10	1	100	1.2
'ahgn08'	1	-8	0	2	1.2
'bhgn08'	1	-8	0	5	1.2
'chgn08'	1	-8	0	10	1.2
'dhgn08'	1	-8	0	100	1.2
'asgn08'	1	-8	1	2	1.2
'bsgn08'	1	-8	1	5	1.2
'csgn08'	1	-8	1	10	1.2
'dsgn08'	1	-8	1	100	1.2
'ahgn06'	1	-6	0	2	1.2
'bhgn06'	1	-6	0	5	1.2
'chgn06'	1	-6	0	10	1.2
'dhgn06'	1	-6	0	100	1.2
'asgn06'	1	-6	1	2	1.2
'bsgn06'	1	-6	1	5	1.2
'csgn06'	1	-6	1	10	1.2
'dsgn06'	1	-6	1	100	1.2
'ahgn05'	1	-5	0	2	1.2
'bhgn05'	1	-5	0	5	1.2

```
'chgn05'   1   -5   0   10    1.2
'dhgn05'   1   -5   0   100   1.2
'asgn05'   1   -5   1   2     1.2
'bsgn05'   1   -5   1   5     1.2
'csgn05'   1   -5   1   10    1.2
'dsgn05'   1   -5   1   100   1.2
'ahgn04'   1   -4   0   2     1.2
'bhgn04'   1   -4   0   5     1.2
'chgn04'   1   -4   0   10    1.2
'dhgn04'   1   -4   0   100   1.2
'asgn04'   1   -4   1   2     1.2
'bsgn04'   1   -4   1   5     1.2
'csgn04'   1   -4   1   10    1.2
'dsgn04'   1   -4   1   100   1.2
'ahgn02'   1   -2   0   2     1.2
'bhgn02'   1   -2   0   5     1.2
'chgn02'   1   -2   0   10    1.2
'dhgn02'   1   -2   0   100   1.2
'asgn02'   1   -2   1   2     1.2
'bsgn02'   1   -2   1   5     1.2
'csgn02'   1   -2   1   10    1.2
'dsgn02'   1   -2   1   100   1.2
'ahgn01'   1   -1   0   2     1.2
'bhgn01'   1   -1   0   5     1.2
'chgn01'   1   -1   0   10    1.2
'dhgn01'   1   -1   0   100   1.2
'asgn01'   1   -1   1   2     1.2
'bsgn01'   1   -1   1   5     1.2
'csgn01'   1   -1   1   10    1.2
'dsgn01'   1   -1   1   100   1.2
'ahgp00'   1   0    0   2     1.2
'bhgp00'   1   0    0   5     1.2
'chgp00'   1   0    0   10    1.2
'dhgp00'   1   0    0   100   1.2
'asgp00'   1   0    1   2     1.2
'bsgp00'   1   0    1   5     1.2
'csgp00'   1   0    1   10    1.2
'dsgp00'   1   0    1   100   1.2
'ahgp01'   1   1    0   2     1.2
'bhgp01'   1   1    0   5     1.2
'chgp01'   1   1    0   10    1.2
'dhgp01'   1   1    0   100   1.2
'asgp01'   1   1    1   2     1.2
'bsgp01'   1   1    1   5     1.2
'csgp01'   1   1    1   10    1.2
'dsgp01'   1   1    1   100   1.2
'ahgp02'   1   2    0   2     1.2
'bhgp02'   1   2    0   5     1.2
'chgp02'   1   2    0   10    1.2
'dhgp02'   1   2    0   100   1.2
'asgp02'   1   2    1   2     1.2
'bsgp02'   1   2    1   5     1.2
'srcc/csgp02.n'  1    2   1     10 1.2
'dsgp02'   1   2    1   100   1.2
'ahgp04'   1   4    0   2     1.2
'bhgp04'   1   4    0   5     1.2
'chgp04'   1   4    0   10    1.2
'dhgp04'   1   4    0   100   1.2
'asgp04'   1   4    1   2     1.2
'bsgp04'   1   4    1   5     1.2
'csgp04'   1   4    1   10    1.2
'dsgp04'   1   4    1   100   1.2
'ahgp05'   1   5    0   2     1.2
'bhgp05'   1   5    0   5     1.2
'chgp05'   1   5    0   10    1.2
'dhgp05'   1   5    0   100   1.2
```

'asgp05'	1	5	1	2	1.2
'bsgp05'	1	5	1	5	1.2
'csgp05'	1	5	1	10	1.2
'dsgp05'	1	5	1	100	1.2
'ahgp06'	1	6	0	2	1.2
'bhgp06'	1	6	0	5	1.2
'chgp06'	1	6	0	10	1.2
'dhgp06'	1	6	0	100	1.2
'asgp06'	1	6	1	2	1.2
'bsgp06'	1	6	1	5	1.2
'csgp06'	1	6	1	10	1.2
'dsgp06'	1	6	1	100	1.2
'ahgp08'	1	8	0	2	1.2
'bhgp08'	1	8	0	5	1.2
'chgp08'	1	8	0	10	1.2
'dhgp08'	1	8	0	100	1.2
'asgp08'	1	8	1	2	1.2
'bsgp08'	1	8	1	5	1.2
'csgp08'	1	8	1	10	1.2
'dsgp08'	1	8	1	100	1.2
'ahgp10'	1	10	0	2	1.2
'bhgp10'	1	10	0	5	1.2
'chgp10'	1	10	0	10	1.2
'dhgp10'	1	10	0	100	1.2
'asgp10'	1	10	1	2	1.2
'bsgp10'	1	10	1	5	1.2
'csgp10'	1	10	1	10	1.2
'dsgp10'	1	10	1	100	1.2
'ahgp12'	1	12	0	2	1.2
'bhgp12'	1	12	0	5	1.2
'chgp12'	1	12	0	10	1.2
'dhgp12'	1	12	0	100	1.2
'asgp12'	1	12	1	2	1.2
'bsgp12'	1	12	1	5	1.2
'csgp12'	1	12	1	10	1.2
'dsgp12'	1	12	1	100	1.2
'ahgp14'	1	14	0	2	1.2
'bhgp14'	1	14	0	5	1.2
'chgp14'	1	14	0	10	1.2
'dhgp14'	1	14	0	100	1.2
'asgp14'	1	14	1	2	1.2
'bsgp14'	1	14	1	5	1.2
'csgp14'	1	14	1	10	1.2
'dsgp14'	1	14	1	100	1.2

APPENDIX D: Sample Output File

```
BEGIN RANGE, FLAT, A, C AND USER
     1.0  133.0  103.3  130.0  139.0
     1.3  133.0  103.3  130.0  138.9
     1.6  133.0  103.2  130.0  138.9
     2.0  132.9  103.2  129.9  138.8
     2.5  132.8  103.1  129.8  138.7
     3.2  132.6  102.9  129.6  138.6
     4.0  132.4  102.7  129.4  138.3
     5.0  132.0  102.3  129.0  138.0
     6.3  131.5  101.8  128.5  137.5
     7.9  130.8  101.1  127.8  136.8
    10.0  129.8  100.1  126.8  135.7
    12.6  128.7   98.9  125.6  134.6
    15.8  127.3   97.5  124.3  133.2
    20.0  125.7   96.0  122.7  131.7
    25.1  123.9   94.2  120.9  129.9
    31.6  122.2   92.4  119.1  128.1
    39.8  120.3   90.6  117.3  126.2
    50.1  118.3   88.5  115.3  124.2
    63.1  116.3   86.6  113.3  122.3
    79.4  114.3   84.5  111.3  120.2
   100.0  112.2   82.5  109.2  118.1
   125.9  110.1   80.4  107.1  116.0
   158.5  107.9   78.2  104.9  113.8
   199.5  105.7   76.0  102.7  111.7
   251.2  103.4   73.7  100.4  109.4
   316.2  101.1   71.4   98.1  107.1
   398.1   98.7   69.0   95.7  104.7
   501.2   96.2   66.5   93.2  102.2
   631.0   93.5   63.8   90.5   99.5
   794.3   90.7   61.0   87.7   96.7
  1000.0   87.7   58.0   84.7   93.7
  1258.9   84.4   54.7   81.4   90.4
  1584.9   80.8   51.1   77.8   86.8
  1995.3   76.7   47.0   73.7   82.7
  2511.9   72.1   42.4   69.1   78.1
  3162.3   66.9   37.2   63.9   72.9
  3981.1   60.8   31.1   57.8   66.8
  5011.9   53.6   23.9   50.6   59.6
  6309.6   51.4   19.6   48.3   57.5
  7943.3   49.3   17.0   46.2   55.5
 10000.0   47.2   14.5   44.0   53.4
 12589.3   45.1   11.8   41.9   51.3
 15848.9   43.0    9.1   39.7   49.3
 19952.6   40.8    6.4   37.4   47.2
```

```
BEGIN RANGE, FLAT, A, C AND USER
   10.0   129.8   100.1   126.8   135.7
 RECEIVER SPECTRUM, 1/3-OCTAVE BANDS 0-43
   25.0    25.0    25.0    25.0    25.0
   95.0    98.0   101.0   104.0   107.0
  110.0   113.0   116.0   119.0   122.0
  125.0   122.0   119.0   116.0   113.0
  110.0   107.0   104.0   101.0    98.0
   95.0    25.0    25.0    25.0    25.0
   25.0    25.0    25.0    25.0    25.0
   25.0    25.0    25.0    25.0    25.0
   25.0    25.0    25.0    25.0
```

ECA Team Distribution

Chief of Engineers
ATTN: CEHEC-IM-LP (2)
ATTN: CEHEC-IM-LH (2)
ATTN: CERD-L
ATTN: CEMP-CE
ATTN: CEMP-EA
ATTN: CEMP-EI (2)
ATTN: CEMP-ZA
ATTN: CEMP-ZM (2)
ATTN: DAIM-EO

HQ USAF/LEEEU 20332

US Army Europe
ODCS/Engineer 09014
ATTN: AEAEN-FE
ATTN: AEAEN-ODCS

AMC 22333
ATTN: AMCEN-A

Fort Belvoir, VA 22060
ATTN: Water Resource Center
ATTN: CECC-R
ATTN: NACEC-FB

Picatinny Arsenal 07801
ATTN: Library

US Military Academy 10996
ATTN: Facilities Engineer
ATTN: Dept of Geography & Environmental Engrng
ATTN: MAEN-A

Naval Air Systems Command 20360
ATTN: Library

Little Rock AFB 72099
ATTN: 314/DEEE

Aberdeen PG, MD 21010
ATTN: Safety Office Range Safety Div
ATTN: US Army Ballistic Res Lab (2)
ATTN: ARNG Operating Activity Ctr
ATTN: Human Engineer Lab

Edgewood Arsenal, MD 21010
ATTN: HSHB-MO-B

NAVFAC 22332
ATTN: Code 2003

Naval Surface Weapons Center 22448
ATTN: N-43

Ft. McPherson, GA 30330
ATTN: AFEN-FEB

US Army Aeromedical Res Lab 36362
ATTN: SGRD-UAS-AS

USAWES 39180
ATTN: WESSEN-B
ATTN: Soils & Pavements Lab
ATTN: C/Structures

Wright-Patterson AFB, OH 45433
ATTN: AAMRL/BB
ATTN: AAMRL/BBE

Ft. Monmouth 07703
ATTN: AMSEL-EW-MD

Dept. of Housing & Urban Dev 20410
ATTN: Environmental Planning Div (2)

Nat'l Institute of Standards & Tech 20899
ATTN: Force & Acoustics Group

Department of Transportation
ATTN: Library 20590

Naval Undersea Center, Code 401 92132

Bureau of National Affairs 20037

Building Research Board 20418

Transportation Research Board 20418

Federal Aviation Administration 20591

AVSCOM 63120-1798
ATTN: SFAE-AV-ASH

Defense Technical Info. Center
ATTN: DTIC-FAB (2)

54
09/94

This publication was reproduced on recycled paper.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1994—3510-S/00076